

Student Name

Student id:

Section #: A

QUESTION ONE: Convert the following C++ code into equivalent assembly code.

{8 pts}

```
int bb[12]={20, 101, 576, 609, 911, 97, 84, 23, 408, 39, 12,69}
cout<< "The mystery numbers are: ";
for(j=0; j<12; j++)
{ if(bb[j] % 2 == 1)
  cout<< bb[j]<< " ";
}
```

cout << endl;

*data*  
bb *word* 20, 101, 576, 609, 911, 97, 84, 23, 408, 39, 12, 69

msg byte "The mystery numbers are: "

*code*

lea edx, msg  
call writestring

mov ecx, lengthof bb  
mov esi, 0

L0: ~~Test esi, 11~~  
mov eax, bb[esi]

~~star~~  
RCL eax, 1  
inc L2

RCL eax, 1  
call writeint

L2: ~~inc~~ add esi, 4

Loop L0

call CRLF

mov al, " "  
call writechar

Student Name

Student id:

Section #: A

QUESTION TWO: Write a sequence of assembly instructions to perform each of the following tasks:

- 1) Give ONE instruction to set bits 4, 5, 14 and 15 in SI register. {1 pt}

1 OR SI, 000030h

- 2) Give ONE instruction to swap the halves in ecx register. {1 pt}

1 ROL ECX, 16

- 3) Give ONE instruction to set the odd-numbered bits in BX to 1. Keep other bits in BX unchanged {1 pt}

1 OR BX, 0AAAAh

- 4) Give ONE instruction to inverse the even-numbered bits in CX. Keep other bits in CX unchanged {1 pt}

1 XOR CX, 5555h

- 5) Divide the signed values 47890CH / C000H {2 pts}

1  
mov eax, 47890Ch  
mov bx, C000h  
idiv bx  
dx:ax = bx

- 6) Give no more than 2 instructions to shift the entire value in DX:AX 1 bit to the left. {2 pts}

2  
shl ax, 1  
rcr dx, 1

- 7) Give no more than 2 instructions to set all bits in the flags register to 1. {2 pts}

2  
~~push~~ pushf  
~~pop~~ popfd

- 8) Give no more than 2 instructions to store in EBX the product of multiplying CL by 1024. CL register may contain any unsigned value. (ADD, MUL and IMUL instructions are not allowed). {2 pts}

2  
movzx ebx, cl  
shl ebx, 10

Student Name

12

Student id:

Section #: A

QUESTION THREE:

{12 pts}

a) MOV AX, 4C9AH  
TEST AX, 0FCD7H  
OR AX, AX

AX = 4C9A H

b) MOV AX, 3AE5H  
MOV BX, 768BH  
ROR BX, 8

BX = 8B76 H

c) MOV AX, 3A90H  
MOV CX, 4CA0H  
IMUL CL

AX = 2A00 H

d) MOV AX, 300BH  
MOV BX, 2640H  
IDIV BL

AX = 0BCC H

e) What will be in registers AX and ESP after executing the following instruction sequence?

MOV ESP, 3000H  
MOV EAX, -1  
MOV ECX, 2C67H  
PUSH ECX  
CMP AX, CX  
JL L9  
PUSH EAX  
AND AX, 3A4CH  
L9: XOR AX, CX

AX = D398 H

ESP = 2FFE H

f) If "JL L9" is replaced by "JB L9", what will be in registers AX, ESP after executing the above code:

AX = 162B H

ESP = 2FFE8 H

And  
1111 1111 1111 1111  
0011 1010 0100 1100  
0011 1010 0100 1100  
XOR 0010 1100 0110 0111  
0001 0110 0010 1011  
1 6 2 B

FFFF  
1111 1111 1111 1111  
XOR 0010 1100 0110 0111  
1101 0011 1001 1000  
D 3 9 8  
2 F F F

Student Name

Student id:

Section #: A

QUESTION FOUR:

{8 pts }

Write a complete assembly program that:

- Prompts the user to enter from the keyboard an integer value in the range 3-50.
- Reads the value and validates it against the above given range limits. If an invalid value is entered, loop until a valid value is entered.
- Calls a procedure TRI that displays a triangle of at signs "@" as shown in the example below
- The procedure TRI accepts as a parameter the entered value in eax register.

For example, if you entered 5 from the keyboard, the output should be the following shape:

```
@@@@@
@@@@@
@@@@
@@@
@@
@
```

```
#include Irvine32.inc
.data
msg byte "enter number from 3-50: ", 0
```

```
code
TRI proc
```

```
mov bl, al
```

```
mov ecx, bl
```

```
mov al, "@"
```

```
call writechar
```

```
loop
```

```
dec bl
```

```
jnz
```

```
ret
```

```
TRI endp
```

```
main proc
```

```
LO: lea edx, msg
```

```
call write string
```

```
call readint
```

```
cmp eax, 3
```

```
jb LO
```

```
cmp eax, 50
```

```
jg LO
```

```
call TRI
```

```
exit
```

```
main endp
```

```
end main
```